International Journal of Computer Science and Application
IJCSA Volume 1, Issue 1, August 2012 PP. 26-30
ISSN(online)
ISSN(print)
www.ij-csa.org

Effect of Tcp Reno Without Maximum Congestion Window in Ring Noc

Mohammad Reza Nouri Rad[1], Teamour Esmaeili[2]
[1 2]Dep.of Computer Engineering DareShahr Branch, Islamic Azad University, Iran
Email: [1]nouri_rad@yahoo.com, [2]esmaeiliteamour@yahoo.com

(Abstract) This paper shows the behavior of TCP Reno with two packet drops without maximum congestion window for increase the reliability of network on chip (NoC).We simulate Ring NoC architecture with Network Simulator 2 (NS2). The simulation results reveal the applicability of TCP Reno protocol in Congestion Control in proposed architecture.
Keywords: Network-on-Chip; TCP Reno; Congestion Window; Ns-2.

1. INTRODUCTION

Packet-based interconnection networks, known as Network-on-Chip (NoC) architectures, are increasingly adopted in System-on-Chip (SoC) designs, which support numerous homogeneous and heterogeneous functional modules. Systems-on-chip (SoCs) for multimedia or telecommunication applications will contain a large number of process-ing elements (PEs) such as a DSP processor, RISC CPU, embedded RAM, graphics engine, etc. As a result, there is a need for high-throughput communications links between these blocks. There exist many bus based SoCs which are widely used in industry such as AMBA [1].

Transmission Control Protocol (TCP) is one of the core protocols of the TCP/IP Protocol Suite. TCP is used to provide reliable data between two nodes and works at the transport layer of the TCP/IP model. TCP operates at a higher level, concerned only with the two end systems, for example, a Web browser and a Web server. In particular, TCP provides reliable, ordered delivery of a stream of bytes from a program on one computer to another program on another computer. Besides the Web, other common applications of TCP include e-mail and file transfer. Among its other management tasks, TCP controls message size, the rate at which messages are exchanged, and network traffic congestion [2].

Commonly used TCP variant is TCP Reno and uses basic AIMD mechanism only to adjust their congestion window size. TCP Reno was the modified version of TCP Tahoe. These protocols are not scalable as the delay-bandwidth product of the network becomes larger [3] because additive increase is too slow and multiple decrease is too fast. Basic TCP uses packet loss only to adjust the congestion window size. So, TCP Vegas and FAST TCP are proposed to cope up the same problem. FAST TCP uses packet loss as well as queuing delay as the congestion control parameter and to adjust window after every RTT (Round Trip Time) [4-7].

TCP/IP model is made up of 4 layers i.e. Application layer, Transport layer, Internet layer, Network layer. Transmission Control Protocol (TCP) is one of the main protocols used at the transport layer. The TCP/IP header is shown as below:

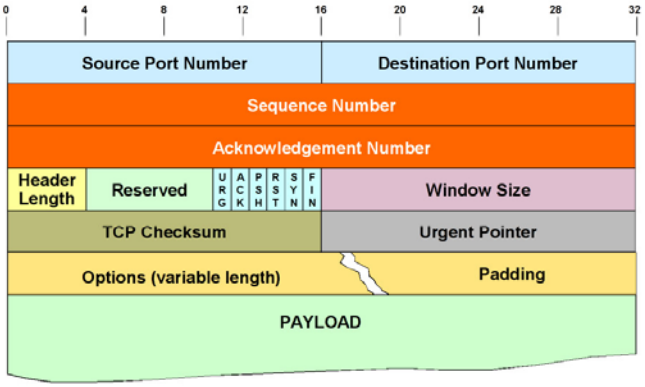

Fig. 1. the TCP header.

Congestion is the phenomenon that occurs at a router when incoming packets arrive at a rate faster than the router can switch (or forward) them to an outgoing link. However, it is important to distinguish contention and congestion. “Contention occurs when multiple packets have to be queued at a switch (or a router) because they are competing for the same output link, whereas congestion means that the switch has so many packets queued that it runs out of buffer space and has to start dropping packets”.

Congestive collapse (or congestion collapse) is a condition which a packet switched computer network can reach, when little or no useful communication is happening due to congestion. Congestion collapse generally occurs at choke points in the network, where the total incoming traffic to a node exceeds the outgoing bandwidth.

2. CLASSIFICATION OF CONGESTION CONTROL ALGORITHMS

The classification of congestion control algorithms in TCP was demonstrate as below:

2.1 Slow Start Algorithm

The Slow Start Algorithm tries to avoid congestion by sending data packets defensively. Therefore, two special variables named congestion window (cwnd) and Slow Start threshold (ssthresh) are stored on sender's side.

Initially, cwnd is sized to one packet when the sender injects a new packet into the network and waits for the Acknowledgment (ACK) [5]. from the receiver. Normally, this packet gets through the network and reaches the recipient in time, so it will be replied by an ACK. If this acknowledgment is received by the sender, cwnd is incremented; if network capacity is reached and packets get lost, the sender does not increment the number of packets any further. That means, by each sending cycle the number of injected data packets is doubled until network's capacity is reached and the required ACK cannot get through. the Slow-Start and Congestion Avoidance was shown s below:

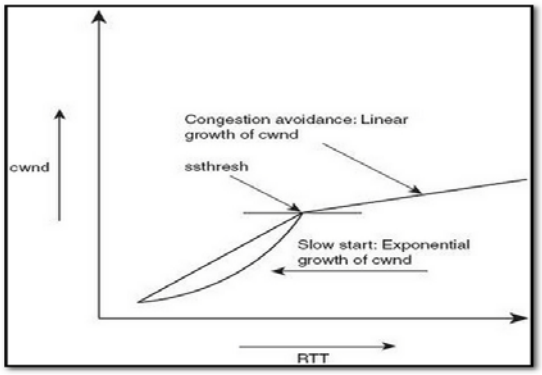

Fig. 2. Slow-Start and Congestion Avoidance.

2.2 Fast Retransmit Algorithm

Fast Retransmit Algorithm uses explicit feedback methods to avoid long timeout periods waiting for packet retransmitting in case of packet loss. Such problems are inherent in packet-switched data networks because every data packet can travel individually through the rest of the network and can use special routes from the sender to the recipient [8]. Consequently, the transmitted data packets will neither reach the recipient in accurate order nor complete continually. Therefore, after detecting a missing packet the recipient sends duplicated ACK packets for the last correct received packet until the missing packet receives. Unfortunately, TCP may use duplicate ACK packets to indicate out-of-order-packets, thus two ACK packets do not necessarily indicate a lost packet. Therefore, if a sender receives multiple ACK packets with the same sequence number, normally at least three of them, these packets indicate the last successfully transmitted packet. the Retransmit Algorithm was shown as below:

2.3 Fast Recovery Algorithm

A special Congestion Avoidance Algorithm often combined with Fast Retransmit to restart transmission at a higher throughput. rate than Slow Start is the FAST Recovery Algorithm. Fast Recovery starts when Fast Retransmit fails to work. If no further duplicate ACK packets are received for Fast Retransmit Algorithm the sender tries to return to normal sending state.

2.4 Congestion Avoidance

Congestion can occur when data arrives on a big pipe (a fast LAN) and gets sent out a smaller pipe (a slower WAN). Congestion can also occur when multiple input streams arrive at a router whose output capacity is less than the sum of the inputs. Congestion avoidance is a way to deal with lost packets [9]. The assumption of the algorithm is that packet loss caused by damage is very small (much less than 1%), therefore the loss of a packet signals congestion somewhere in the network between the source and destination. There are two indications of packet loss:

A timeout occurring.

Receipt of duplicate ACKs.

Congestion avoidance and slow start are independent algorithms with different objectives. But when congestion occurs TCP must slow down its transmission rate of packets into the network, and then invoke slow start to get things going again. In practice they are implemented together. Congestion avoidance and slow start require that two variables be maintained for each connection: a congestion window(cwnd), and a slow start threshold size(ssthresh).

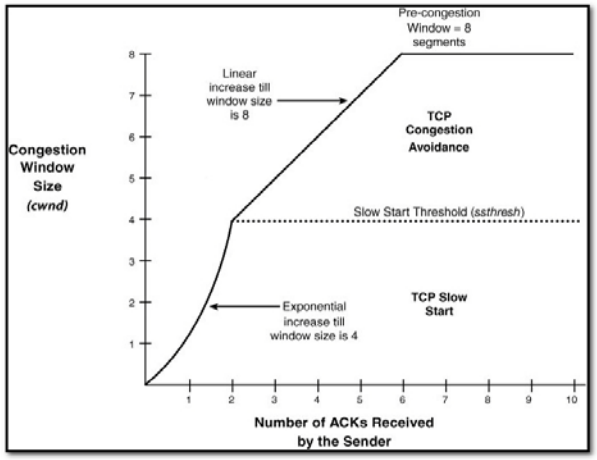

Fig. 3. Fast Retransmit Algorithm.

3. TYPES OF TCP PROTOCOLS

3.1 Packet Loss-based Protocols

These are the protocols which uses packet drop probability as the main factor for adjusting the window size. These variants of TCP use congestion control algorithms. There were developed initially and are still used. Loss based TCP protocols are more aggressive than the delay based TCP protocols [6]. These are classified as below:

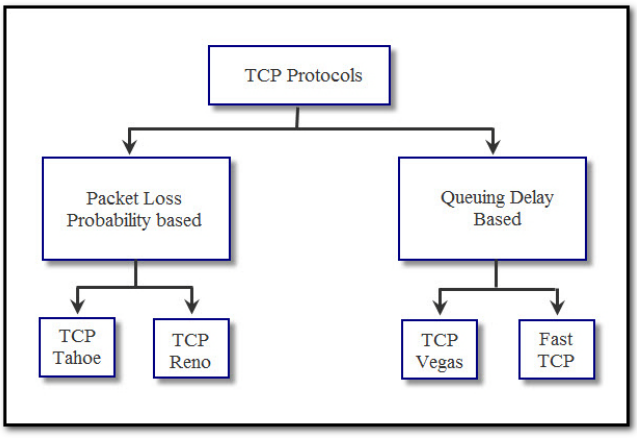

Fig. 4. The Types of TCP Protocols.

1) TCP Tahoe

TCP Tahoe is the TCP variant developed by Jacobson in 1988 [10]. It uses Additive Increase Multiplicative Decrease (AIMD) algorithm to adjust window size. It means that increases the congestion window by one for successful packet delivery and reduces the window to half of its actual size in case of data loss or any delay only when it receives the first negative acknowledge. In case of timeout event, it reduces congestion window to 1 maximum segment size (MSS) [11]. TCP Tahoe has following specification:

• TCP Tahoe uses packet loss probability to adjust the congestion window size.

• During Slow Start stage, TCP Tahoe increases window size exponentially i.e. for every acknowledgement received, it sends two packets.

• During Congestion Avoidance, it increases the window size by one packet per Round Trip Time (RTT) so as to avoid congestion.

• In case of packet loss, it reduces the window size to one and enters in Slow Start stage.

2) TCP Reno

This Reno retains the basic principle of Tahoe, such as slow starts and the coarse grain re-transmit timer. However it adds some intelligence over it so that lost packets are detected earlier and the pipeline is not emptied every time a packet is lost. Reno requires that we receive immediate acknowledgement whenever a segment is received. The logic behind this is that whenever we receive a duplicate acknowledgment, then his duplicate acknowledgment could have been received if the next segment in sequence expected, has been delayed in the network and the segments reached there out of order or else that the packet is lost. If we receive a number of duplicate acknowledgements then that means that sufficient time has passed and even if the segment had taken a longer path, it should have gotten to the receiver by now. There is a very high probability that it was lost so Reno suggests an algorithm called 'Fast Re-Transmit'. Whenever we receive 3 duplicate ACK's we take it as a sign that the segment was lost, so we re-transmit the segment without waiting for timeout. Thus we manage to re-transmit the segment with the pipe almost full [12].

3.2 Delay-Based TCP Protocols

Delay-based algorithms were developed so as to provide stable throughput at the receiver end. These TCP variants use congestion avoidance algorithms to avoid the packet loss and are less aggressive than packet loss based TCP protocols. Delay-based algorithms can maintain a constant window size avoiding the oscillations inherent in loss-based algorithms [13].These are classified as below:

3) TCP Vegas

TCP Vegas is a congestion control or network congestion avoidance algorithm that emphasizes packet delay, rather than packet loss, to determine the rate at which to send packets. TCP Vegas detects congestion during every stage based on increasing Round Trip Time (RTT) values of the packets in the connection unlike Reno, Tahoe etc. which detect congestion only after it has actually happened via packet drops [14].

• TCP Vegas adjusts the source rate before actually packet is dropped.

• Queuing delay is the difference between base RTT and avg RTT.

• TCP Vegas decreases the source rate in case of increase in queuing delay value and increases in case of decrease in queuing delay.

4) FAST TCP

FAST TCP is a new TCP congestion control algorithm for high-speed long-distance networks; it aims to rapidly stabilize networks into steady, efficient and fair operating points. It uses queuing delay, in addition to packet loss, as a congestion signal. Queuing delay provides a finer measure of congestion and scales more naturally with network capacity than packet loss probability does [15]. Using the queuing delay as a congestion measure in its window updating equation [4], allows FAST TCP to overcome difficulties [16] encountered by currently used algorithms (such as TCP Reno [12]) in networks with large bandwidth-delay products.

4. SYSTEM ARCHITECTURE

4.1 Hardware Architectures

The common characteristic of NoC architectures is that the constituent IP cores communicate with each other through switches [17]. In network-on-chip the bandwidth between resource (IPs) and switches is very higher than the bandwidth between switch to switch. The ring NoC has shown as below:

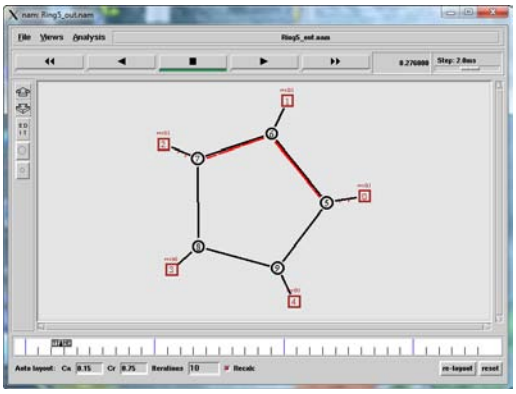

Fig. 5. The Ring Network on Chip.

As shown in fig. 5, The Square was processor elements and the circle elements were switches that connect to each other.

5. EVALUATIONS

5.1 Simulation Framework

In this paper, we have modeled our NoC architecture concepts with the widely used network simulator ns-2 [18]. NS-2 has been widely applied in research related to the design and evaluation of computer networks and to evaluate various design options for NoC architectures [19], including the design of routers, communication protocols, etc.

We consider that two Res11 and Res44 communicate with each other through network. the bandwidth between resources and switches was equals to 8 megabits/sec and the bandwidth between switches was equals to 800 kilobit/sec. the link delay between resources and switches was equals to 10 milliseconds and the link delay between switches was equals to 15 milliseconds. The ssthresh is set two packets.

5.2 Fast Retransmission

As shown in fig. 6, TCP Reno cannot recover from multiple losses in one window size without a timeout. The first packet loss can be recovered by fast retransmission. The second packet loss cannot get enough duplicate acknowledgments for fast retransmission and has to wait for retransmission timeout to recover from loosing.

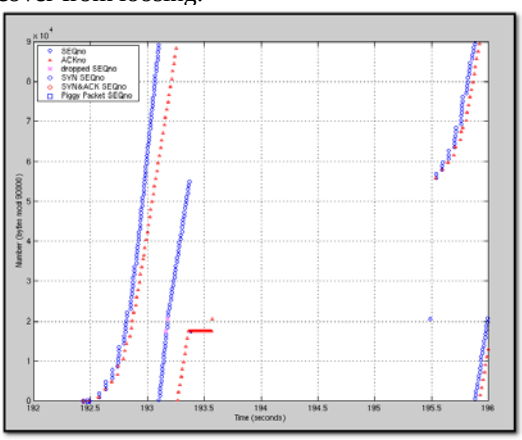

Fig. 6. Sequence Number of TCP Reno in switch.

5.3 Congestion Window

As shown in fig. 7, Congestion window is the flow control set by the traffic source. Advertised window is the flow control performed by the TCP sink. Thus maximum congestion window has the same effect as the TCP sink advertised window size.

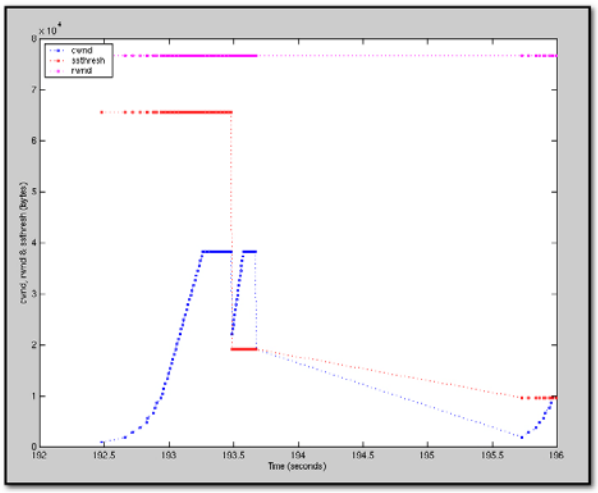

Fig. 7. Congestion window (cwnd), slow start threshold (ssthresh) and receiver window (rwnd).

6. CONCLUSION AND FUTUREWORK

This paper shows that TCP Tahoe protocol have the most influence on congestion control. In the future, we will investigate the problem of optimal buffer allocation for loss-based and delay-based protocols in more detail. Also, we will explore mechanisms for detecting loss-based flows sharing the same link queue with delay-based flows.

References

[1] J. Park, I. Kim, S. Kim, S. Park, B. Koo, K. Shin, K. Seo, and J. Cha,
"MPEG-4 video codec on an ARM core and AMBA," in Proc. of Workshop and Exhibition on MPEG-4, Jun. 2001, pp. 95–98.

[2] http://en.wikipedia.org/wiki/Transmission_Control_Protocol.

[3] Cheng Peng Fu, Bin Zhou, Jian Ling Zhang, "Modeling TCP Veno Throughput over Wired/Wireless Networks," IEEE COMMUNICATIONS LETTERS, VOL. 11 NO. 9, SEPTEMBER 2007.

[4] C. Jin et al., "FAST TCP: From Theory to Experiments," IEEE Network, vol. 19, no. 1, pp. 4– 11, Jan./Feb. 2005.

[5] J. Wang, D. X. Wei, and S. H. Low, "Modelling and Stability of FAST TCP," in Proc. IEEE INFOCOM 2005, Miami, FL, Mar. 2005.

[6] C. Jin, D. Wei, and S. H. Low, "FAST TCP for high-speed long-distance networks," Internet draft draft-jwl-tcp-fast-01.txt. [Online]. http://netlab.caltech.edu/pub/papers/draft-jwl-tcp-fast-01.txt.

[7] David X., Wei Cheng Jin, Steven H. Low Sanjay Hegde, “FAST TCP: Motivation, Architecture, Algorithms, Performance,” IEEE/ACM Transactions on Networking, 14(6):1246-1259, Dec 2006.

[8] J. Wang, A. Tang, S. H. Low,“Local stability of Fast TCP”, proc. IEEE conf. decision and control, December 2004.

[9] Jacobson, V., "Congestion Avoidance and Control Computer Communication Review, vol. 18, no. 4, pp. 314-329, ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z", August 1988.

[10] T. V. Lakshman, Member, IEEE, and Upamanyu Madhow, Senior Member, IEEE, “The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss,” IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 5, NO. 3, JUNE 1997.

[11] By Steven H. Low, Fernando Paganini, and John C. Doyle, “Internet Congestion Control,” IEEE Control Systems Magazine, 0272-1708/02/$17.00©2002 IEEE.

[12] Mandakini Tayade et al, " REVIEW OF DIFFERENT TCP VARIANTS IN AD-HOC NETWORKS", International Journal of Engineering Scince and Technology(IJEST), Vol. 3, No. 3, March 2011.

[13] T.V. Lakshman, Upamanyu Madhow, Bernhard Suter, “TCP/IP Performance with Random Loss and Bidirectional Congestion”, IEEE/ACM Transactions on networking, Vol. 8, NO. 5, 2000.

[14] Cheng Peng Fu, Bin Zhou, Jian Ling Zhang,“Modeling TCP Veno Throughput over Wired/Wireless Networks”, IEEE communications letters, Vol. 11, No. 9, 2007.

[15] C. Jin, D. Wei, and S. H. Low, “FAST TCP: Motivation, architecture, algorithms, performance,” in Proc. IEEE INFOCOM 2004, vol. 4, Mar. 2004, pp. 2490-2501.

[16] F. Paganini, Z. Wang, J. C. Doyle, and S. H. Low, “Congestion control for high performance, stability, and fairness in general networks,” IEEE/ACM Trans. Networking, vol. 13, pp. 43-56, Feb. 2005.

[17] Nostrum, http://www.imit.kth.se/info/FOFU/Nostrum.

[18] www.isi.edu/nsnam/ns.

[19] R. Lemaire, F. Clermidy, Y. Durand, D. Lattard, and A. Jerraya, “Performance Evaluation of a NoC-Based Design for MC-CDMA Telecommunications Using NS-2,” in The 16th IEEE International Workshop on Rapid System Prototyping, Jun. 2005, pp. 24–30.